MEMORY BUS ANALYSIS FOR 6502 MICROPROCESSOR SYSTEMS

A Project

Presented

to the Faculty of

California State University, Chico

In Partial Fulfillment

of the Requirements for the Degree

Master of Arts

in

Special Major: Microprocessor Systems

---

by

James D. Moore

Spring 1983

APPROVED BY THE DEAN OF THE GRADUATE SCHOOL:

John S. Morgan

APPROVED BY THE GRADUATE ADVISORY COMMITTEE:

Byron M. Jackson
Graduate Advisor

H.R. Luxenberg, Chairman

John Young

Robert Maurer

TABLE OF CONTENTS

LIST OF FIGURES

PAGE

FIGURE

CHAPTER I

INTRODUCTION

There exists in the world of microcomputers a great need to see more of what is "going on" within a particular system. This is often necessary for problem solving, instructional purposes, program design, and even the satisfaction of one's own curiosity. Many programs are also "protected" so that even casual analysis is quite impossible. Program step and trace functions are frequently available as software tools but are rendered useless in a great many circumstances by protection schemes.

The purpose of this project was to develop a simple and inexpensive hardware device to assist programmers in gaining access to the inner workings of both hardware and software. This device is commonly known as a Bus Rider when found in development system application.

Information on this type of hardware is practically non-existent. A literature review revealed only two display-type program aids for microcomputer systems, neither of which alters the processor speed. The Display Board and Bug Lights circuit are useful but rather limited in their

ability to display information.[1] Both use a series of discrete LEDs to indicate data, which necessitates a constant hexidecimal conversion on the part of the programmer. They are also particular about when and what they display.

Microcomputer systems normally perform several hundred thousand operations per second and must therefore be slowed down a great deal to make any sense of the data transfer at the bus level. This is possible with the 6502 microprocessor because it is an early design and was required to handle the slow memory devices of the time.

The device resulting from this project is specifically designed for popular 6502 based microcomputer systems like the Apple II. It consists of a synchronized clock circuit that pulses the Ready Line of the microprocessor and provides variable speed, slow motion, and pause capabilities anywhere within an operating program. A set of six hexadecimal LED displays then decode and display memory locations and the data found in them.

A program can be run in "slow motion" and one can actually watch the processor access specific memory locations to transfer data. In essence it allows one to examine, in great detail, the most intimate operations of a microcomputer and its programs.

---

[1] John Bell Engineering, Inc., 1982 Summer Catalog (San Carlos, 1982); Marvin L. De Jong, Programming and Interfacing the 6502, with Experiments (Indianapolis: Howard W. Sams & Co., Inc., 1981).

CHAPTER II

CIRCUIT DESIGN

Functional Analysis

The Bus Rider was designed to control the Ready Line of the 6502 microprocessor. This allows the operator to choose a speed that facilitates, in real time, exploration of the system. Figure 1 shows a complete schematic of the Bus Rider, and descriptions of the various circuit functions follow.

Clock Section

The 6502 Ready Line is normally held high, and when pulled low will cause the processor to halt following an instruction cycle. The active low signal is provided by IC 1, a 556 timer, under the control of a 1 megohm potentiometer.

IC 1 has two sections which are configured differently. The first section is designed to be a continuous output oscillator whose frequency is dependent on the value of R1 and C1. Switch S1 is used to alter the value of C1 for the various speed ranges. A satisfactory frequency range for exploring graphics is 150 Hz to 20kHZ.

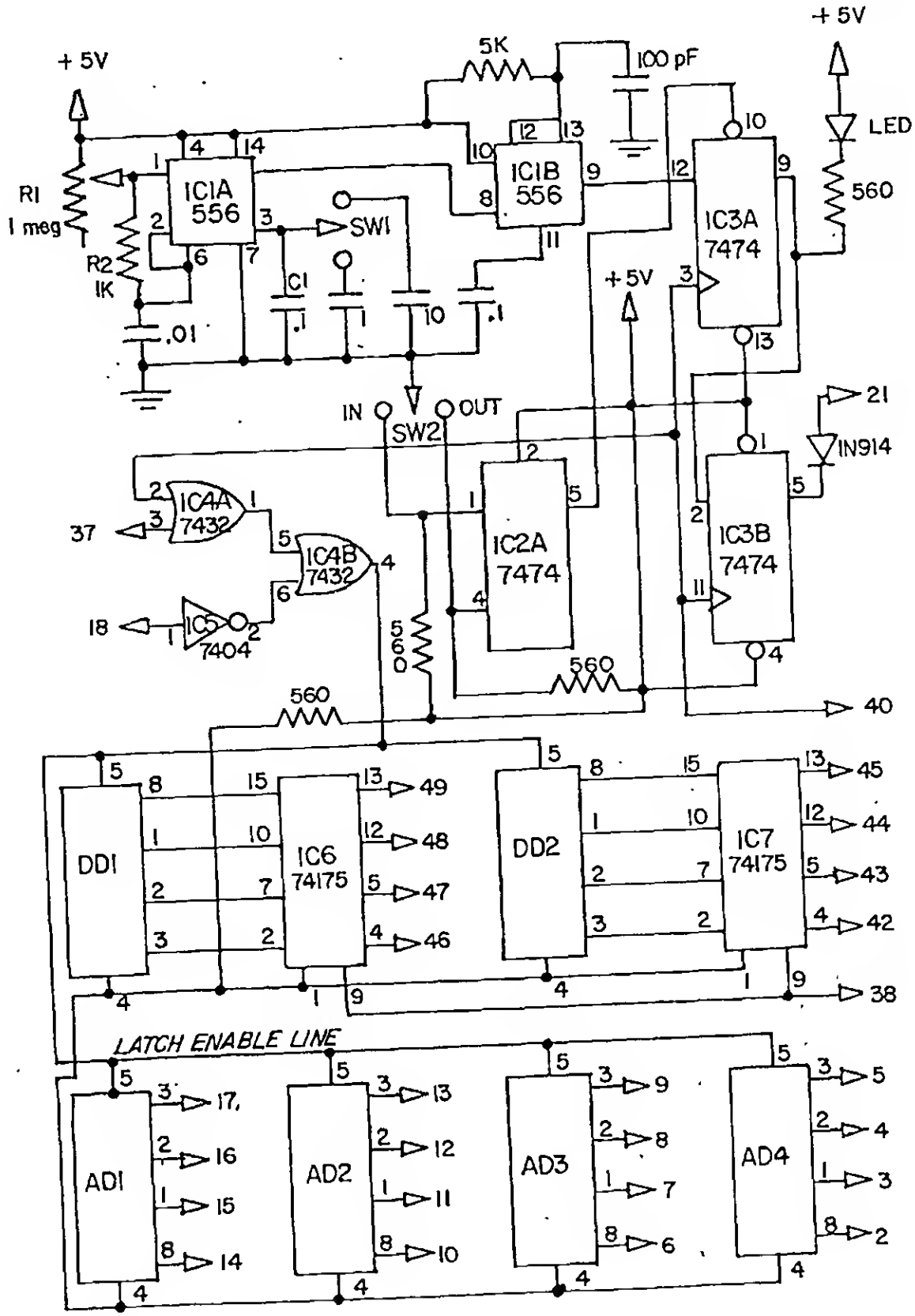


Figure 1

Bus Rider Schematic

Analysis of displayed data requires a much lower range of about .15Hz to 10kHz. The formula for frequency is:

$$F = \frac{1.44}{(R1 + 2R2)C1}$$

The second section of IC 1 is a monostable one-shot configuration. It squares up the oscillator output and gives a controlled pulse width.

Obviously the ratio of low to high conditions, or duty cycle, on the Ready Line will control the processor speed. With a very small duty cycle (less than 1%) low frequencies will provide longer periods of a low condition.

Latch Section

IC 2a, a 7474 D flip flop, provides the on/off toggle control for the clock circuit. It functions as a latch and engages the unit on command. It also disengages the circuit for a "reset" or "off" condition.

Sync Section

Ic 3, another dual D flip flop, is used to synchronize the pulse generator output with the system clock. The first section controls the out put to the Ready Line. The second section clocks the final output at the trailing edge of the phase zero clock, satisfying the 6502 timing requirements. A 1N914 diode is used for isolation.

Display Section

The display section is fairly simple due to the selection of integrated readouts. The HP 5082-7340

hexidecimal indicators have on-board decoders and drivers with memory. They decode positive 8-4-2-1 logic inputs into the 16 unique conditions of 0-9 and A-F. They also have a line to blank the displays, which is handy when the displays are changing too rapidly to be of use.

Four of the displays are directly connected to the address bus lines (pins 2-17) of the expansion slot. The two data displays are wired to the data bus through flip flops for latching and buffering purposes. Valid data is available for a much shorter period on the data bus and reliability is increased by the additional latching.

Timing

Accurate display of information depends entirely upon timing. The displays themselves contain a quad latch capable of storing 4 line BCD input. The data to be displayed must be available to the input 50ns or more before the enable rises to a logic "1" level and should be held for 50ns or more after the enable rises to ensure reliable operation. This 50ns setup and hold time allows data to be clocked into a display array at up to 10 mHz.

On the system side, the address on the address bus is valid about 300ns after the phase 1 clock goes high. It remains valid until the phase zero clock goes low. Valid data is put onto the bus about 300ns after the phase 1 clock and the read/write line go low as illustrated in figure 2.

The Apple phase 1 clock (at 1mHz) is brought to an "OR" network (IC 6) with an inverted read/write line and the Q3 general purpose timing signal. The unique set of conditions when these three signals are low places the display enable command at the 300ns window for reliable data, as indicated on the timing diagram.

Construction

Expansion Slot

Figure 3 illustrates the Apple expansion slot pinout. All of the lines necessary for the operation of the Bus Rider are available at these peripheral connectors. The lines include the phase 1 clock (pin 30), the phase zero clock (pin 40), Q3 general purpose timing signal (pin 37), the Ready Line (pin 21), the read/write line (pin 18), the address bus (pins 2-17), the data bus (pins 42-29), ground (pin 26), +5 volts (pin 25), and reset (pin 31).

Prototype Card

All components except the display devices and controls were mounted on a prototype board of standard Apple configuration. 0.1 uf disk capacitors were used to decouple power supply voltages at the rate of one for each four integrated circuits. A 50 pin header was used to connect cabling from the prototype board to the control and display

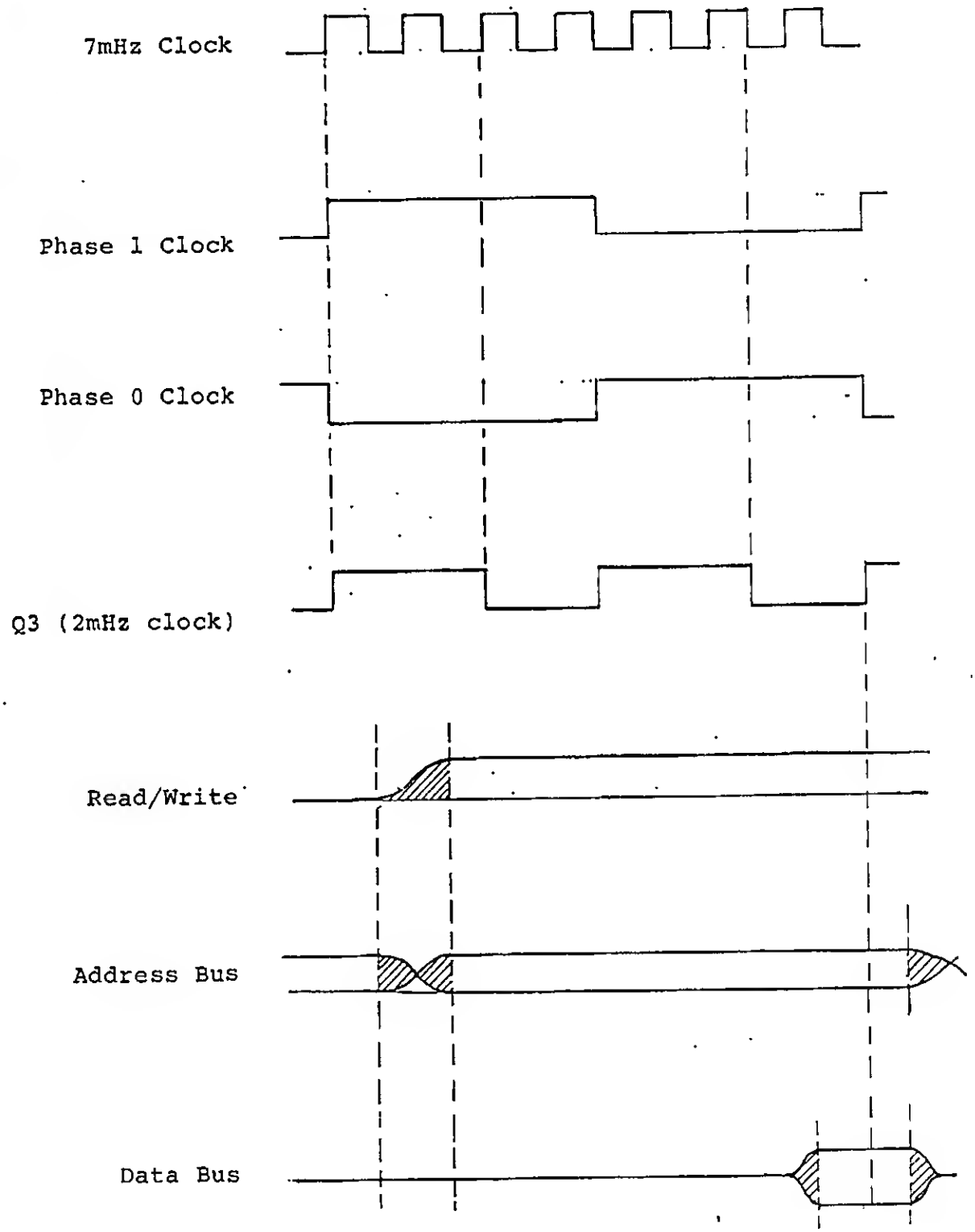


Figure 2

Timing Diagram

	Ground	26	- -	25	+5V	
	Dma In	27	- -	24	DMA Out	
	INT In	28	- -	23	INT Out	
	NMI	29	- -	22	DMA	
	IRQ	30	- -	21	Ready Line	
	RESET	31	- -	20	I/O Strobe	
	INH	32	- -	19	N.C.	
	-12V	33	- -	18	Read/Write	
	-5V	34	- -	17	A15	
	N.C.	35	- -	16	A14	
	7 mHz Clock	36	- -	15	A13	
	Q3 (2mHz clock)	37	- -	14	A12	
	Phase 1 Clock	38	- -	13	A11	
	USER 1	39	- -	12	A10	
	Phase 0 Clock	40	- -	11	A9	
	Device Select	41	- -	10	A8	Address Bus
		D7	42	- -	9	A7
		D6	43	- -	8	A6
		D5	44	- -	7	A5
		D4	45	- -	6	A4
Data Bus		D3	46	- -	5	A3
		D2	47	- -	4	A2
		D1	48	- -	3	A1
		D0	49	- -	2	A0
		+12v	50	- -	1	I/O Select

Figure 3

Apple Expansion Slot Pinout

enclosure. The 1.6 ns/ft propagation delay of the PVC ribbon cable used was negligible compared to the 300ns timing windows available.

Enclosure

The display enclosure contains four hexidecimal indicators for address display, two hexidecimal indicators for data display, the speed control, range switch, blanking switch, engage and disengage buttons. Photographs of the circuit board and enclosure are provided in appendix B.

Future Developments

Universal Interfacing

The Bus Rider device could be made universal for most 6502 microcomputer systems by the use of a 40 pin IC interface clip. This would attach directly to the microprocessor chip and bring the necessary lines to an appropriate enclosure containing all circuitry and display components. In this way the interface is no longer specialized for a particular expansion slot as found on the Apple.

Additional Circuitry

Another design improvement could be the addition of a counter so that a program could be run and automatically stopped at increments of say 100 steps or so. A halt function under program control allowing breakpoints to be set for debugging would also be useful. Firmware

intelligence contained in the device would give a still further measure of versatility and control. This package approach could be geared toward the programmer, the student, the hardware designer or trouble-shooter as an addition to the computer system.

CHAPTER III

CIRCUIT APPLICATIONS

Instructional Uses

Instructional applications for the Bus Rider are obvious. Simple machine language routines can be run, stopped, corrected and examined in much greater detail than was previously possible. The instant feedback available from the display would take much of the drudgery out of learning assembly language programming. The device is also useful for the exploration of memory allocation and Hi-Res screen mapping.

Other possible applications would include debugging programs. By slowing down the microprocessor as the problem point approaches, one can visually see the sequence of addresses and data leading up to the failure. Endless loops and unwanted jumps, normally very difficult problems, are particularly susceptible to Bus Rider analysis. System problems, those involving hardware, are also directly approachable using a stop/display technique.

Perhaps the most useful education for a programmer is to explore the most useful program, BASIC. All the tools for data and memory management are contained in BASIC and are made available at a fundamental level with the Bus

Rider. Professional programs in and of themselves are excellent learning aids for the programmer. Again, BASIC may be first among these.

Graphics

Programs involving graphics are probably the most interesting and intriguing aspect of microcomputer use today, and they account in large measure for the popularity of machines like the Apple. Most notable are "arcade style" games whose smooth, multicolored imagery is nothing short of miraculous considering the multitude of Apple idiosyncrasies and memory allocation limitations. These types of programs are especially informative for the way in which they manage memory, run in real-time, and interact with the user.

The most instructive examples of graphics type programming are found in the best programs, which are generally access-protected. The fact that the Apple operating DOS resides as software gives opportunity for a variety of protection schemes which modify DOS protocols and structure. Systems employing firmware/hardware disk control do not allow for this kind of flexibility. Perhaps this accounts for the great number of quality programs available for the Apple. The protection schemes themselves are an impressive learning tool. The Bus Rider makes it possible to "see into" these programs as they run and analyze them on several different levels.

Control of the 6502 clock rate with complex graphics displays creates an effective "slow motion" screen action. One can see how screen objects move and behave as they are drawn, erased and redrawn. How fast screen objects move in relation to each other when different shapes are updated is made clear through the bus activity. It is also possible to run a game at reduced speeds to give a player more reaction time to see how the program logic develops. Most modern games are sophisticated pieces of programming and a great deal can be learned by analyzing their construction.

Memory Allocation

When using the Bus Rider address/data display, a whole new window into the internal structure of programs is made available. Typical "arcade style" games will plot their images to the Hi-Res screen area, which on the Apple is one of two 8k blocks. The most common scheme for graphics management is that of Block Graphics. This technique involves the use of a library of shapes in free RAM that are loaded as needed to locations in screen memory. It makes extreme memory space demands as the price for high speed action. Newer games use an overlay technique from disk to overcome RAM limitations in the Apple.

What is not obvious from the literature or normal programming activity is the structure of the Apple Hi-Res screen and the limitations this imposes on the programmer. At the bit level we find that only seven bits per byte are

involved in screen display. The overall resolution in 280 horizontal by 192 vertical dots, a 1.4:1 ratio. The eighth bit serves as a toggle for color pairs displayed by the lower seven bits. A further complication is that given color pairs can occupy only even or odd lines on the screen. There are four colors available paired as follows:

	even	odd
high	blue	orange
low	violet	green

Two or more adjacent dots on a horizontal line give the color white. Color conflicts can occur with two unpaired colors within a byte, when scrolling across even/odd lines or with the mixing of "high and low" black or white.

The screen itself is mapped in a very convoluted manner. It consists of three groups of 64 horizontal lines each of which contain 40 bytes. Each group in turn contains eight groups of eight lines. Any Hi-Res action therefore involves a conversion formula to deal with the actual screen mapping. Lastly, since a primary and secondary Hi-Res area are available separate management schemes are frequently used provide flicker free movement. See appendix A for a general memory allocation map for the Apple.

CHAPTER IV

CONCLUSIONS

The development of the Bus Rider circuitry, physical construction and subsequent troubleshooting phase involved a great deal of learning about microprocessor systems. Many other fields of study such as transmission line theory and EMI shielding had to be investigated during the construction process. In many cases the contributed effects could be ignored considering the speeds and risetimes involved in present microcomputer technology. It was, however, necessary to consider a great many system factors and weigh them accordingly. Very delicate timing and voltage threshold windows had to be adjusted for reliable device performance.

The many and varied applications of the Bus Rider open up completely new avenues of investigation for small computer systems. Problems need to be assessed from many different angles and this device provides a unique window into the heart of a microprocessor.

The Bus Rider is useful for a wide variety of applications including program analysis, isolating subroutines, understanding memory allocation, system

diagnosis, assorted instructional purposes and variable speed graphics. It is a powerful tool for guiding creativity and imagination in the microprocessor field.

BIBLIOGRAPHY

BIBLIOGRAPHY

Apple Computer Inc., Apple II Reference Manual, Apple Computer Inc., 1981

Apple Computer Inc., "Hobby/Prototyping Board", Apple Computer Inc., 1982

De Jong, Marvin L., Programming & Interfacing the 6502 With Experiments, Howard W. Sams & Co., 1980

Dougherty, William E., The Apple II Monitor Peeled, Apple Computer Inc., 1981

Elcomp, Microcomputer Hardware Handbook, Ing. W Hofacker GmbH, 1982

Hewlett Packard Co., "Technical Data Sheet for Hexadecimal Indicators Series 5082-7300", Hewlett Packard Inc., 1973

Hewlett-Packard Co., "Application Note 934", Hewlett Packard Co., 1974

John Bell Engineering, 1982 Catalog, John Bell Engineering, 1982

Lancaster, Don, TTL Cookbook, Howard W. Sams & Co., 1974

Leventhal, Lance, 6502 Assembly Language Programming Osborne/McGraw-Hill, 1979

Luebbert, William F., What's Where in the Apple, Micro Ink, 1982

Mims, Forrest M. III, Engineers Notebook, Radio Shack, 1979

APPENDIX A

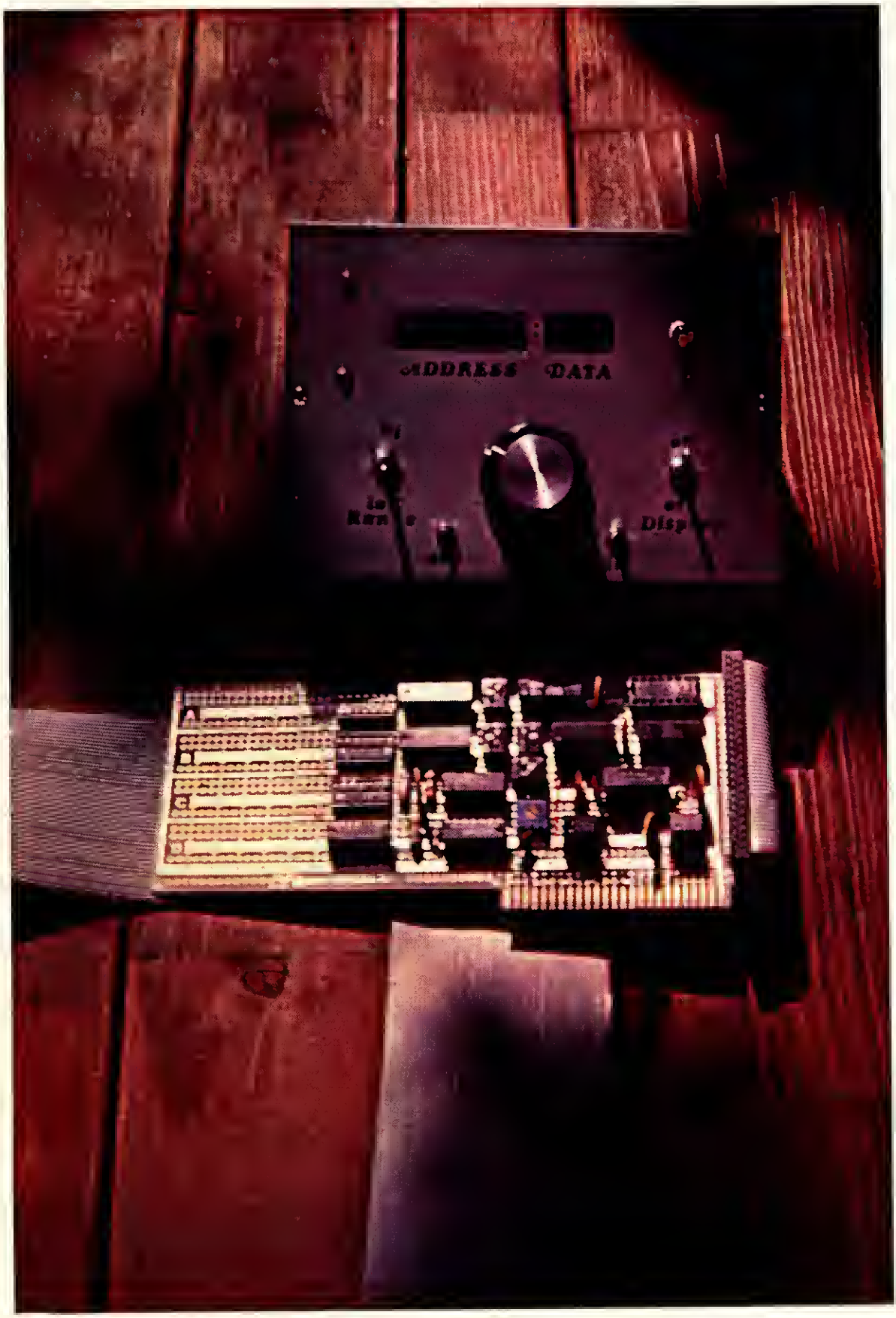
APPLE II MEMORY ALLOCATION

2K	Monitor	$F800 - $FFFF	
10K	FP Basic	$D000 - $F7FF	Extra Memory Bank Switched over 12k
4K	I/O	$C000 - $CFFF	
10K	DOS	$9600 - $BFFF	
14K	User RAM	$6000 - $9600	
8K	Hi-Res Screen 2	$4000 - $5FFF	
8K	Hi-Res Screen 1	$2000 - $3FFF	
6K	User RAM	$0800 - $1FFF	
2K	System	$0000 - $07FF	(see below)

page 0: Zero Page
page 1: Stack
page 2: Input Buffer
page 3: User Utility
pg.4-7: Output Buffer

APPENDIX B


Apple II


ADDRESS
DATA
Disp


ADDRESS
DATA
hi
low
Range
out
Speed
in
on
off
Display